

19. (New) The program storage device of claim 1, wherein the allowing a type declaration includes allowing a type declaration in a programming language interpreter to be embedded within an object identifier declaration.

20. (New) The program storage device of claim 1, wherein the type declaration allows the programming language to operate on an object declared in the type declaration without an explicit call to construct the object.

21. (New) The program storage device of claim 1, wherein the type declaration allows the programming language to automatically instantiate an object being declared in the type declaration when the type declaration embedded with the object identifier declaration is first read by the programming language.

REMARKS

Reconsideration of the application is respectfully requested. In the Office Action, corrected drawings were required. In response, corrected Figure 1 is being submitted with this amendment. The disclosure was objected to because of several informalities. The specification is being amended to correct the informalities noted. With regard to the rejection of claims 4-10 under 35 U.S.C. §112, second paragraph, the parenthetical expressions are deleted. With regard to the rejection of claim 1-15 under 35 U.S.C. §101, the claims are amended to recite a computer program storage device, which is a statutory patentable subject matter. Accordingly, these objections and rejections raised with respect to the drawings, disclosure, and claims are believed to be overcome.

Also in the Office Action, claims 1, 2, 11-14, and 16-18 were rejected under 35 U.S.C. §102(b) as allegedly being anticipated by Charles Simonyi, "Program identifier Naming Conventions" ("Simonyi"). Claims 1, 7, and 9 were rejected under 35 U.S.C. §102(b) as allegedly being anticipated by T. Berners-Lee et al. "Universal Resource Identifier (URI): Generic Syntax." ("Berners-Lee"). Claims 3-6 were rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Simonyi in view of J.J. Kuslich, "Writing Database Applications with Enterprise Server 3.0; Part II: Strategies & Techniques" ("Kuslich"). Claim 10 was rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Berners-Lee. Claim 8 was rejected under 35 U.S.C. §103(a)

as allegedly being unpatentable over Simonyi in view of "Microsoft Visual Basic 5.0 Programmer's Guide" ("VB5"). Claim 15 was rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Simonyi in view of Roger Jack, "Top Ten ASP Tips," Microsoft Interactive Developer ("Jack").

With respect to Simonyi and Kuslich articles, although the Office Action includes dates for these articles, there appear to be no published dates specified in the copies provided. Accordingly, Applicant respectfully requests copies that have the publication dates printed on the article in order to ascertain that these articles indeed qualify as prior art. Further, even if the articles do predate the filing date of the present application, it is submitted that neither Simonyi nor Kuslich anticipates the claims in the present application as amended for at least the following reasons.

With respect to the rejection under 35 U.S.C. §102(b) over Simonyi, it is submitted that Simonyi does not disclose or suggest every element claimed in independent claims 1, 16, and 17. Simonyi appears to disclose a naming convention for variable names, in which a variable name includes a type of a variable followed by a qualifier. This naming convention, Simonyi states on page 3, lines 12-14, is "for the benefit of humans ... Capabilities or restrictions of the programming environment are not at issue here. The exact determination of what constitutes a 'type' is not critical, either."

On the other hand, claim 1 in the present application recites in part, "allowing a type declaration in a programming language to be embedded within an object identifier declaration." According to claim 1, a type declaration, for example in a programming language, may be embedded within an object identifier declaration because of a unique syntax provided in the programming language. The programming language is then able to interpret this syntax. Simonyi does not disclose or suggest to declare a type within an identifier. Rather Simonyi suggests how to name a variable. With Simonyi's naming convention, an additional type declaration is further necessary to complete the type declaration in a programming language. Similarly, Simonyi does not disclose or suggest to "declare an object type in a programming language ... , wherein the object identifier name is interpreted by a machine as having the object type indicator" as claimed in independent claims 16 and 17. To reiterate, Simonyi simply suggests a naming convention for variables in a source code for human readability. Without more,

a programming language receiving a variable name as suggested by Simonyi would not be able to interpret or process the variable name. For at least these reasons, it is submitted that claims 1, 16, and 17 and their respective dependent claims are patentable over Simonyi.

With regard to the rejection of claims 1, 7, and 9 under 35 U.S.C. §102(b) over Berners-Lee, Berners-Lee appears to disclose an Internet standard protocol for accessing desired locations. On the other hand, claims 1, 7, and 9 recite "allowing a type declaration in a programming language to be embedded within an object identifier declaration." Unlike the protocol scheme disclosed in Berners-Lee, the type declaration claimed in claims 1, 7, and 9 may include any type declaration, including any object types not limited by a protocol scheme only, and may be interpreted by a programming language. Accordingly, it is submitted that Berners-Lee does not disclose every element claimed in claims 1, 7, and 9, and therefore, for at least the foregoing reasons, claims 1, 7, and 9 are patentable over Berners-Lee.

With regard to the rejection of claims 3-6 under 35 U.S.C. §103(a), Kuslich also appears to disclose a variable naming convention. Like Simonyi, Kuslich does not disclose or suggest to embed a type declaration in a programming language within an object identifier declaration as claimed in claims 3-6. Accordingly, for at least the foregoing reason, it is submitted that claims 3-6 are patentable over Simonyi and Kuslich.

With regard to the rejection of claim 10 under 35 U.S.C. §103(a), it is submitted that Berners-Lee as discussed above does not disclose or suggest embedding a type declaration in a programming language within an object identifier declaration. Accordingly, for at least the foregoing reason, it is submitted that claim 10 is patentable over Berners-Lee.

With regard to the rejection of claim 8, it is submitted that claim 8 is patentable over Simonyi and VB5, at least for the reasons discussed above with reference to the Simonyi rejections and because VB5 does not disclose or suggest what Simonyi fails to disclose or suggest.

Similarly, with regard to claim 15, it is submitted that claim 15 is patentable over Simonyi and Jack, at least for the reasons discussed above with reference to the Simonyi rejections and because Jack does not disclose or suggest what Simonyi fails to

disclose or suggest.

Further, new claims 19-21 are being added in this amendment. The support for the new claims are found in the specification. Applicant believes the new claims are also patentable over the cited references.

For the foregoing reasons, it is submitted that all pending claims in this application are patentable over the cited references. This communication is believed to be fully responsive to the Office Action and every effort has been made to place the application in condition for allowance. Attached hereto is a marked-up version of the changes made to the specification by the current amendment. The attached page is captioned **"Version with markings to show changes made."**

If a telephone interview would be of assistance in advancing prosecution of the subject application, Examiner is requested to telephone the number provided below.

Respectfully submitted,

Date: February 19, 2003


Eunhee Park
Registration No. 42,976
Baker & McKenzie
805 Third Avenue
New York, NY 10022
Telephone (212) 751-5700
Facsimile (212) 759-9133

VERSION WITH MARKINGS TO SHOW CHANGES MADE**IN THE DRAWINGS:**

Figure 1 in permanent dark ink is filed herewith.

IN THE SPECIFICATION:

Paragraph beginning at line 3 of page 2 has been amended as follows:

The presently prevalent scripting languages, PERL and JavaTM, are very powerful, but are also very complex to use and very resource-intensive to accomplish even the simplest of tasks. Therefore, it is highly desirable to have a programming language syntax that is simple to implement and yet efficiently cover most of what developers need to create high-end interactive applications.

Paragraph beginning at line 8 of page 3 has been amended as follows:

Moreover, because the type-declarations are embedded with the object names, thereby rendering the objects self-contained, the code including type declaration of the present invention can be easily embedded and/or ported into a code of another language, for example, hypertext markup language ("HTML"). The present invention can also be easily integrated into an interactive development environment ("IDE") such as Visual BasicTM and JavaTM Symantic Café.

Paragraph beginning at line 7 of page 4 has been amended as follows:

The present invention is directed to a programming language syntax that embeds object type declaration in the object name. The objects are self-documenting because

its object type is embedded in each object. The present invention is related to a co-pending U.S. Patent Application Serial No. 09/583,673 (Attorney Docket No. 56129050-4) entitled DYNAMIC OBJECT SYNTHESIS WITH AUTOMATIC LATE BINDING, filed on May 31, 2000, the disclosure of which is incorporated herein in its entirety by reference thereto.

Paragraph beginning at line 4 of page 7 has been amended as follows:

1. Syntax

<% DynaScript %>

These markers delineate a DynaScript™ code within an HTML document. This can occur more than once in a document, wherever dynamic content is needed.

Paragraph beginning at line 4 of page 20 has been amended as follows:

The alert function causes a message to be displayed within a standard JavaScript™ dialog box. The dialog remains until the user responds by clicking on the "OK" button. An example usage of the alert function is: alert ("An error has occurred\n" + "Please click on OK below to continue");

Paragraph beginning at line 25 of page 28 has been amended as follows:

~~When~~ When only a single connection is necessary, a simple syntax using the default connection and cursor object may be used. In this case, the syntax appears more like traditional C-like function calls. ~~Figure 2 illustrates an example.~~ Another feature of the single connection approach is that SQL statements may be inserted into the script directly and need not be enclosed within a method call, e.g., prepare(). The bindparam,

execute, and fetch methods of the default cursor may be invoked without an object declaration.

Paragraph beginning at line 3 of page 29 has been amended as follows:

Multiple connections may be created and managed simultaneously. Each connection may also have multiple result sets returned, which may be managed with cursors. ~~Figure 2 illustrates an example of code using the multiple connection.~~

IN THE CLAIMS:

Please add claims 19-21.

Please amend claims 1-18 as follows:

1. (Once Amended) ~~An object type declaration syntax comprising:~~

~~an object identifier;~~

~~an object type indicator indicating a type of an object identified by the object identifier; and~~

~~a join attribute associating the object identifier with the object type indicator.~~

A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing an object type declaration syntax, comprising:

allowing a type declaration in a programming language to be embedded within an object identifier declaration; and

allowing the type declaration to be delimited from the object identifier declaration using a joint attribute.

2. (Once Amended) ~~The object type declaration syntax as claimed in claim 1,~~
~~wherein the object type indicator is embedded as part of the object identifier.~~ program storage device of claim 1, wherein the allowing a type declaration includes allowing a type declaration in a programming language compiler to be embedded within an object identifier declaration.

3. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes a database object type.

4. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes a SQL ("SQL") database object type.

5. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes a connection ("CONN") database object type.

6. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes a cursor ("CURSOR") database object type.

7. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes a universal resource locator ("URL") object type.

8. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes an environment ("ENV") object type.

9. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes a hypertext markup language ("HTML") object type.

10. (Once Amended) The ~~object type declaration syntax as claimed in~~ program storage device of claim 1, wherein the ~~object type~~ declaration indicator includes an extensible markup language ("XML") object type.

11. (Once Amended) The ~~object type-declaration syntax as claimed in~~ program storage device of claim 1, wherein the joint attribute is concatenated to the object type declaration indicator.

12. (Once Amended) The ~~object type-declaration syntax as claimed in~~ program storage device of claim 11, wherein the object identifier declaration is concatenated to the joint attribute.

13. (Once Amended) The ~~object type-declaration syntax as claimed in~~ program storage device of claim 1, wherein the joint attribute is concatenated to the object identifier declaration.

14. (Once Amended) The ~~object type-declaration syntax as claimed in~~ program storage device of claim 13, wherein the ~~object type~~ declaration indicator is concatenated to the joint attribute.

15. (Once Amended) The object type-declaration syntax as claimed in claim 1, wherein the object identifier declaration includes dynamically evaluated expressions.

16. (Once Amended) A method of declaring an object type in a programming language, comprising:

embedding an object type indicator with an object identifier name, wherein the object identifier name is interpreted by a machine as having the object type indicator.

17. (Unchanged) A method of declaring an object type in a programming language, comprising:

prepending an object type indicator with an object identifier name, wherein the object identifier name is interpreted by a machine as having the object type indicator.

18. (Unchanged) The method of declaring an object type in a programming language as claimed in claim 16, wherein the step of embedding includes:

joining the object type indicator with the object identifier name with a joint symbol.

19. (New) The program storage device of claim 1, wherein the allowing a type declaration includes allowing a type declaration in a programming language interpreter to be embedded within an object identifier declaration.

20. (New) The program storage device of claim 1, wherein the type declaration allows the programming language to operate on an object declared in the type declaration without an explicit call to construct the object.

21. (New) The program storage device of claim 1, wherein the type declaration allows the programming language to automatically instantiate an object being declared in the type declaration when the type declaration embedded with the object identifier declaration is first read by the programming language.